TagStack:Automated System for Predicting Tags in StackOverflow

Sonam Sonam, Ayushi Verma
Computer Science and Engineering
Jaypee Institute of Information Technology
Noida, India
sonammansa@gmail.com, ayushiv26@gmail.com

Sangeeta Lal, Neetu Sardana
Computer Science and Engineering
Jaypee Institute of Information Technology
Noida, India
sangeeta@jiit.ac.in, sardana.neetu@gmail.com

Abstract—**Stack Overflow is one of the popular online media for the programmers to share their knowledge and experience. Developers seeks help from online communities such as Stack Overflow. The objective of our research is to ease the tagging of questions on Stack Overflow. This work proposes TagStack system, a machine learning and feedback-based framework for predicting tags on Stack Overflow. We perform experiment on real world and publically available dataset, and results shows that TagStack system is effective in predicting tags on Stack Overflow.**

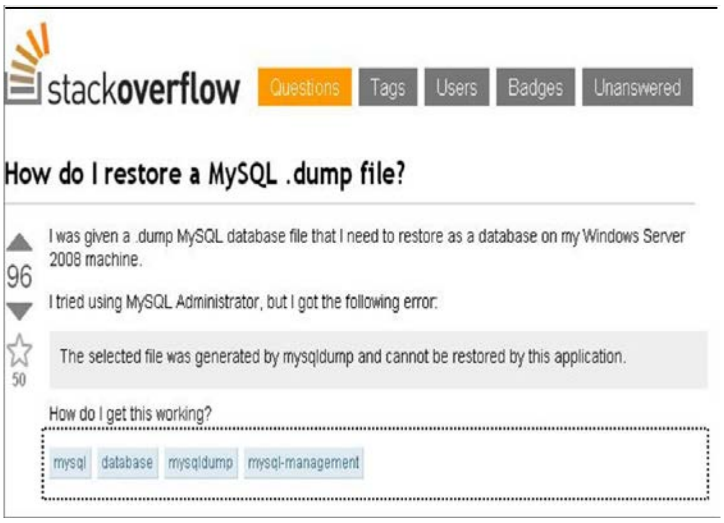
Keywords—Tag;Automate;StackOverflow;Q&A; Reccomendation; Cosine Similarity

I. Introduction

Complexity and size of software is increasing day by day and everyday some new programming language or tool is coming into the market. Software development involves mixture of diverse tools, platforms and technologies. Due to which it is necessary for software practitioners to update and widen their skill sets according to the existing software development trend. Recently, there is growth of many online communities for software development, which not only help software developers in adapting new technologies but also provide them support for their software related issues. These online communities are mainly divided into three categories: Code sharing sites (github.com), Question-Answer (Q&A) based communities (stackoverflow.com) and Tutorial sites (w3schools.com). Online Q&A communities like peakinternet[1], codeproject[2], superuser[3], StackOverflow[4](SO) etc. provide answers to user queries and reduces the burden of developers by providing them similar type of examples and bug fixing codes. SO is one of the largest Q&A websites and it is broadly utilized by software developers seeking programming help globally.

[1]http://www.peakinternet.com/
[2]http://www.codeproject.com/
[3]http://www.superuser.com/
[4]http://www.stackoverflow.com/
[5]https://github.com/
[6]http://www.w3schools.com/

Quick response time and high-quality answers are few reasons of SO popularity. SO uses a reputation-based system where a member is awarded points for each correct answer andmembers with high reputation are also awarded special privileges. Any user can become member of SO community and can ask development related questions. SO, provides easy to use interface where a developer can ask question by giving simple description of the problem. When a user posts a question on SO, he needs to provide the relevant tags to the posted question. Tags are mandatory field of SO interface and user cannot post question without tags. A user is required to give at least three tags. Tags need to be closely associated with the question that the user is asking. Figure-1 shows example of a questions asked on SO and the tags associated with the question.



How do I restore a MySQL .dump file?

I was given a .dump MySQL database file that I need to restore as a database on my Windows Server 2008 machine.

I tried using MySQL Administrator, but I got the following error:

The selected file was generated by mysqldump and cannot be restored by this application.

How do I get this working?

Fig.1. Snapshot of the Stack Overflow Site

Question in Figure 1 is related to MySQL database and hence tags which are related to MySQL database (mysql, database, mysql-dump, mysql-management) are attached with the question.

Tags are very useful for the SO website as they help in better categorization of the questions. Categorization of the questions is beneficial for the users as well for the SO experts in following way:

- **Similar questions detection:** Tags can be used to find similar questions at the time of posting question. Tags can also be used by community experts while searching for a potential question that experts would like to answer.
- **Type of questions:** Tags can be used to categorize question domain i.e. to which programming language or technology they belong to. This information can be beneficial for software development community and can be used to answer many quantitative and qualitative questions, like which language is more popular? Or which API is confusing developers?

When user post a question on SO, he faces difficulty in identifying correct set of tags for the question. Some users just select terms in the question descriptions as the tags, but these terms might not represent the most important features of the question. There can be some hidden tags as well which are not present in the text of the question [1]. Due to which sometimes user assign incorrect tags to the question. To illustrate following are the comments of the one of the users on SO, where he describes the problem of identifying correct set of tags [19]:

*" **HOW CAN I DETERMINE A SUITABLE TAG FOR MY QUESTION ON STACK OVERFLOW?***
I have asked some questions on Stack Overflow and added tags to them, but others always seem to re-edit or add tags to them anyway. I think that means I made a mistake or was confused when I chose tags for my question."

This shows the difficulty of a user in identifying correct set of tags. Hence, automated tag recommendation system can be beneficial for the users for identifying correct set of tags for their questions. Recently automated tag recommendation problem is receiving lots of attention for software engineering research community (discussed in related work). This work presents a model - "TagStack"- for predicting tags on Stack Overflow. We also present results of the proposed system on real world dataset. Following are research contribution of this paper:

- This paper presents TagStack- an automated system for predicting tags on SO websites. TagStack system uses a machine learning and feedback-based mechanism for predicting tags on SO.
- This paper presents results of applying TagStack system on real world, publically available dataset. Results shows that TagStack system is effective in predicting tags.

II. RELATED WORK

In this section, we have discussed the related work with respect to predictions on SO. We have categorized related work in various domains and presented closely related work under different domains.

A. Predictions on Tags

Xin et al. [1] worked on SO & Free code information websites and proposed an algorithm "TagCombine". Author considered three components- Multilabel ranking, similarity-based ranking, tag term-based ranking. Zhang et al. [13] told that the correlations between different labels should be exploited to facilitate multilabel learning.

Zangerle et al. [12] gave the tag recommendation to standardize folksonomies in microblogging settings. Hash tag recommendations are used while creating a new tweet.

Bakshy et al. [15] did the research work on measuring the impact on twitter. To compute the impact scores for a given URL post, the author has traced the distribution of the URL from the source at a node over a series of retweets by the user's followers.

Treude et al. [9] has worked on describing, how the tagging helps to link social and technical aspects in software development. Author has used aspects such as frequency of tag used, most frequently used tags, number of tag users to work on.

Xu et al. [10] presented the measures for tagging system with high coverage of multiple facades,popularity and minimum efforts. Authors characterized tags to content-based tags, context-based tags etc. and used a probabilistic method to recommend tags.

Lipczak et al. [4] used collaborative tagging on other websites and used two main approaches – Graph based approach & Content based approach. Our paper presents "TagStack" system a novel approach to predict tags on SO using machine learning based framework.

B. Predictions on Closed Questions

On SO a question is marked closed if it is not relevant for the site or creates redundancy. Correa et al. [2] used label propagation method to forecast the quality of questions on SO with some features – favorite votes, question score, closure time, answering pattern, question status etc to find out the best accuracy. Lezina et al. [7] used Vowpal Wabbit (VW), support vector machine and random forest for comparison between the accuracies. VW emphases on the method for examples to an operationallearning algorithm.

C. Predictions on Unanswered questions

Asaduzzaman et al. [14] did predictions on unanswered questions. Author used two classifiers- Random Forest & J48 classifier. The author found the features of unreciprocated questions to identify reasons for why questions are not answered. Authors reported that questions are not answered if they belong to one of the following:"Too short", "unclear", "vague", " hard to follow", "program specific without a program snippet or proper explanation", "too hard", "too specific",impatient", " irregular or inconsiderate members","answering their own question", "does not have any answer", "duplicate question", "answer no longer relevant or needed", "fails to attract an expert member" etc.

D. Predictions on Deleted Questions

Correa et al. [16] presented work on characterization and predicts the deletion at the time of creation of questions. Authors discussed about – why questions are deleted, who deletes the questions etc. Prediction of deleted question is outlined as the task of binary classification. Authors presented results of classification on real world dataset. Authors used 47 features constructed on profile, community, content of question and text composition for their task prediction like duration of account, number of previous questions, average answer score, average favorite votes, number of URLs, length of code snippets, total characters in the question text, total number of words in title, average title word length etc.

III. Preliminaries

This subsection describes previous knowledge required to understand the work presented in the paper.

TF-IDF (Term Frequency–Inverse Document Frequency)[5]: It is a numerical statistic which shows the importance of a word in a document or corpus. It is frequently used as a weighting feature in retrieval and mining of data.

$$\boxed{\text{Tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)} \qquad (1)$$

Following are the details of the symbols used in the above equation, where:

t = number of terms

d = each document in the collection of the documents

D = collection of documents.

In the above equation, To find the inverse document frequency (idf(t,D)) of t in d, take log of total number of documents (D). We use this method to calculate the similarity between questions on the basis of their text by calculating the tf-idf scores of all questions using gensim library in python. Then, top 10 similar questions and their tags are extracted. It is also used to give tf.idf scores to features (i.e. words) extracted from the title and body for closed question prediction.

Naïve bayes classification[5]: It is a probabilistic classifier formed on applying Bayes' theorem with naïve independence conventions. We use this procedure for further classification in tags recommended by tf-idf. Then we recommend top 10 tags classified by this algorithm. Importance of selection Naïve bayes is because it is a good method for classification and predictor. Naïve bayes is very simple and stable method.[17]Naive Bayesian to the document classification problem: Consider that documents (i.e., questions) are found from a number of document's classes which can be shown as set of tags. The probability of the i-th tag of a document which lies in a document from class C is written as p(t_i|C). Hence, the probability of document D containing all the tags t_i of class C, is shown as:

$$\boxed{p(D|C) = \prod_{i=1}^{i=10} p(t_i|C)} \qquad (2)$$

Now the question is: "what is the probability that a given document D belongs to a given class C", i.e. p(C|D).
By explanation,

$$\boxed{\begin{aligned} p(D|C) &= \frac{p(D \cap C)}{p(C)} \\ p(C|D) &= \frac{P(C \cap D)}{p(D)} \end{aligned}} \qquad (3)$$

Bayes' theorem works on these into probability statement in relation to likelihood.

$$\boxed{\begin{aligned} p(C|D) &= \left(\frac{p(C)}{p(D)}\right) * p(D|C) \\ &= \frac{p(C)}{p(D)} * \prod_{i=1}^{i=10} p(t_i|C) \end{aligned}} \qquad (4)$$

IV. TagStack System

This section describes details of TagStack system proposed in this paper.

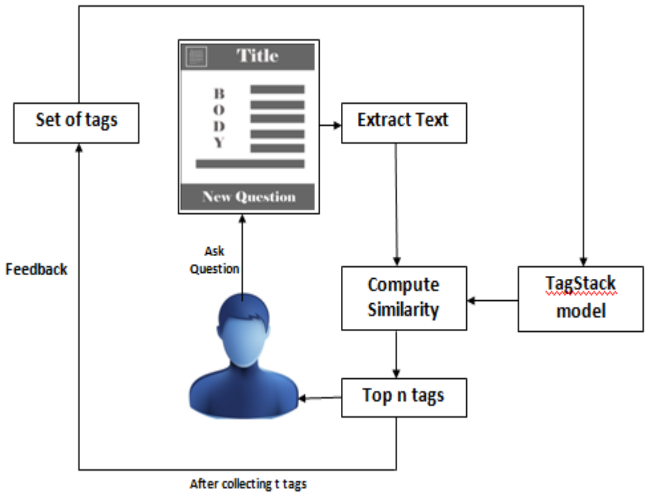


Fig.2. Scenario of TagStack System

A. System Details

TagStack system is an automated system for recommending tags. TagStack system predicts the top n tags for a given question using a classifier. Figure 2 shows overview of TagStack system. TagStack system has mainly three parts- model building, recommendation and feedback. TagStack system extracts features from the question content such as title and body present in the training dataset and builds a model based on Naïve Bayesian classifier. Naïve Bayesian is used for classification because it is found to effective in text classification in earlier researches. After recommending tags system input then as feedback for next iterations.

B. Dataset

SO delivers all user created content on stack exchange website. We used the data till May 2013, table 1 describes other details of the dataset. Firstly, data is extracted from xml file by using xml.sax library which is time efficient as well as uses less memory. HTML tags and stop words are removed from the data. Title and body are extracted from the question and tf-idf (described in section III-C) score is calculated for each question.

Table 2
Motivating examples showing importance of title and body for tag prediction task

Id	Title	Body	Tags
4	While setting a **forms opacity** should I use a decimal or double?	I am new to **C#** and wants to use a track-bar to change a **forms opacity**. Here is my code: decimal trans = trackBar1.Value / 5000; this. **Opacity** = trans; When I try to build it I get this error: Cannot implicitly convert **type** decimal to double.	**c#**, winforms, **forms**, type-conversion, **opacity**
6	Why doesn't the percentage width child in absolutely positioned parent work in **IE 7**	When I use a percentage-based width on the child div it collapses to 0 widths on **IE7** but not on Firefox or Safari. Is there an easy fix for this besides the pixel-based width on the child? Is there an area of the **CSS** specification that covers this?	html, **css**, css3, **internet explorer-7**
8	Tool for Converting **Visual J#** code to **C#**	Are there any conversion tools for **porting** from **Visual J#** code to C#?	**c#**, code-generation, j#, **visualj# , Port**
9	How do I calculate someone's age	Given a **DateTime** showing a person's birthday. How should I calculate the age?	c#, .net, **datetime**

Table 1
Dataset used to evaluate TagStack system

	Field	Value
1.	Users	2.36 M (Registered Users)
2.	Questions in Training set	2000 (Random)
3.	Questions in Testing set	1000 (Random)
4.	Tags predicted per Question	10
5.	Start date of data	July 2008
6.	End date of data	May 2013

C. Model building and Recommendation

This section describes the model building procedure of TagStack system. Detailed model of TagStack system has been shown in the figure 3.

Feature extraction and Bag-of-Word (BOW): First step of TagStack system model building is feature extraction. TagStack system extracts title and body of the posted question. After initial preprocessing TagStack system splits it on spaces and converts it to BOW representation. Tf-idf are then calculated for each question.

Finding Tf-idf score: Tf-idf score increases when the terrn frequency increases in document. We found top 10 questions or posts out set of random 2000 entries of database.

Classifier learning: Tf-idf scores of each question with its respective questions are then used to train Naïve Bayesian classifier.

Training Naïve Bayes classifier: Naïve Bayesian classifier is applied on top n posts which are suggested by Tf-idf score (where n=10) to predict top 10 tags.

Feedback: After every t iterations the output of TagStack system is sent as feedback into the system, to improve the accuracy of the classifier. In this work we have set value of t as 50.

V. Results

This section describes results obtained by implementing TagStack system on real world dataset. Accuracy (equation-5) is used to measure the effectiveness of the TagStack system. Accuracy is calculated in the following way:

$$\boxed{Accuracy = \frac{(t_p + t_n)}{(t_p + f_p + t_n + f_n)}} \quad (5)$$

With respect to our works true positives (t_p) are the tags which are relevant and should be retrieved. Tags which are not relevant and also not retrieved are true negatives (t_n). Tagswhich are retrieved but not relevant are false positives (f_p). False negatives (f_n) are the tags which are not retrieved but are relevant.

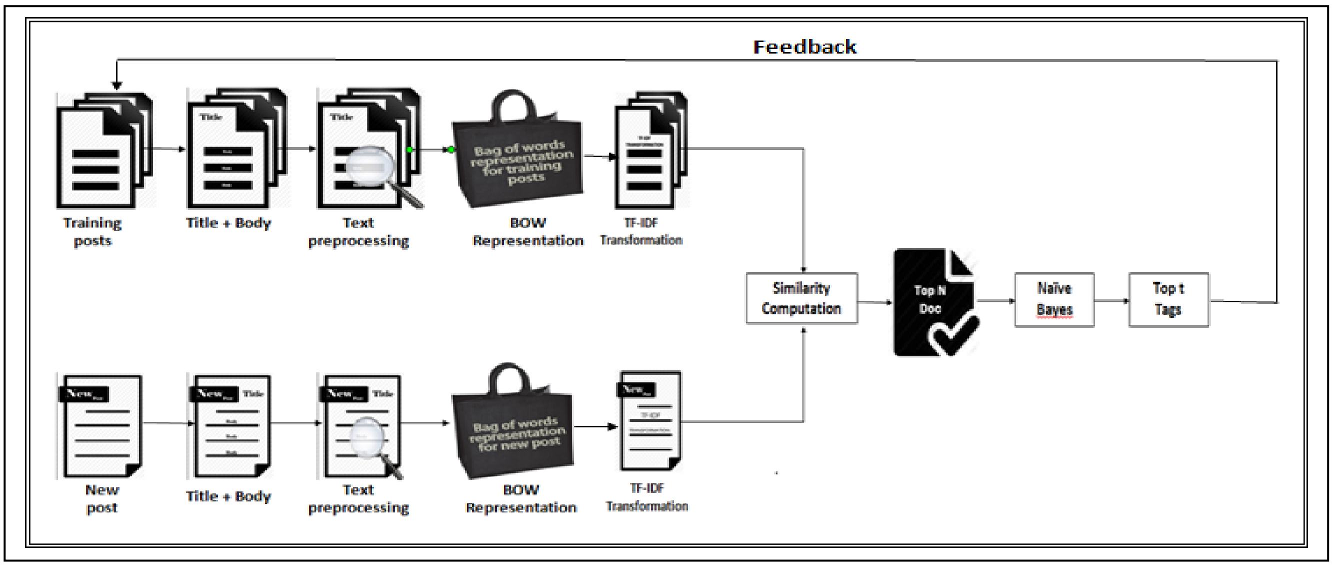


Fig.3. Flow Diagram of TagStack System

This work shows results TagStack system for both scenarios, simple and with feedback. Table 3 shows accuracy value obtained using TagStack system. Table 3 shows improvements obtained in result when feedback is applied in the system. Hence shows effectiveness of our model. Accuracy value before using feedback is around 98% which is increased to 99% when feedback is used in the system, refer table 3 and figure 4. As in table-3, we see that how recommended tags are different from actual given tags and also tags varies from "without feedback" system to "with feedback" system. Tags are aform of metadata which helps in identifying the type of Questions in StackOverflow.

- Figure 5 and Figure 6 shows output of TagStack system by varying number of tags recommended, from top1 to top5. Accuracy values are slightly decreasing may because of increase in true negatives or false positives. Figure-5 for accuracies of tags corresponding to each post by taking top 1,2...n tags at a time(where n=5 for our TagStack system). Green triangles show the accuracies of posts if system recommends top1 tag. Red for 2 tags, blue for3 tags, yellow for 4 tags, cyan stars for recommending 5 tags are shown in figure-5.
- Figure-6 shows accuracies corresponding to full data that how it's falling when count of tag recommendation increases. Red line shows the flow of falling accuracies as number of tags recommended is increasing. Blue marks showing the exact point oftheiraccuracies.

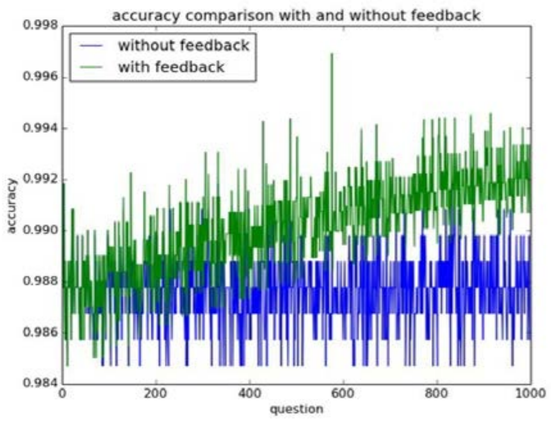


Fig.4. Comparison of accuracies in TagStack

We found some results for TagStack system like: accuracies for recommending top 1-5 tags. Here are two figures:

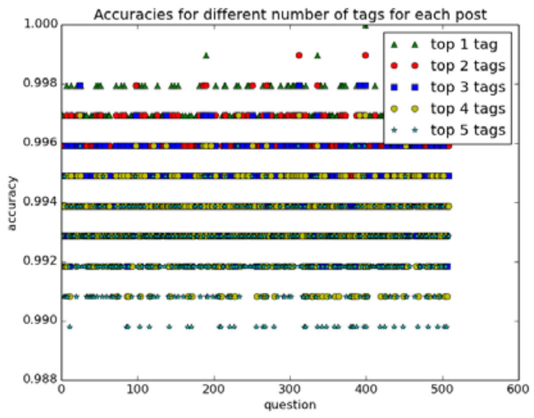


Fig.5. Represents accuracies corresponding to each post for the recommendation

Table 3
Tags Recommended

Doc id	Tags Recommended				Actual tags given
	Without Feedback		With Feedback		
	Tags	Accuracy	Tags	Accuracy	
27899	reflection, error-handling, usercontrols, adobe, macromedia, flash, xml, python, web-services, master-pages	0.9877	monads, parsing, reflection, unix, search, python, adobe, flash, macromedia, winforms	0.9920	amazon-s3,amazon-web-services
27910	math, payment, master-pages, asp.net, .net, asp.net-2.0, data-structures, file, binary, language-agnostic	0.9877	math, payment, opengl, data-structures, file, binary, language-agnostic, image-processing, artificial-intelligence, c#	0.9920	regex,doi
27916	bdd, .net, database, documentation, data-dictionary, c++, visual-studio, sql-server-2005, paging, c#	0.9877	.net, testing, bdd, data-dictionary, database, documentation, sql-server, sql-server-2005, paging, c#	0.9933	sql-server,unit-testing
27921	jbutton, java, swing, image, storage, blob, theory, database, usersgroup, reporting-services	0.9887	delphi, vcl, components, mindmapping, keyboard, mouse, desktop, asp.net, windows-server-2003, printing	0.9926	asp.net,image,thumbnails
27928	puzzle, datetime, parallel-processing, crc32, c#, .net, regex, parsing, colors, language-agnostic	0.9857	puzzle, datetime, crc32, video, audio, mp3, c#, boost-graph, c++, colors	0.9906	math,maps,mapping,latitude-longitude
27931	osx, textwrangler, linux, java, csv, data-conversion, visual-studio, tfs, python, xelement	0.9867	linux, fogbugz, bug-tracking, ontime, tortoisesvn, visualsvn-server, visual-studio, tfs, python, linq	0.9913	xml,xslt,namespaces

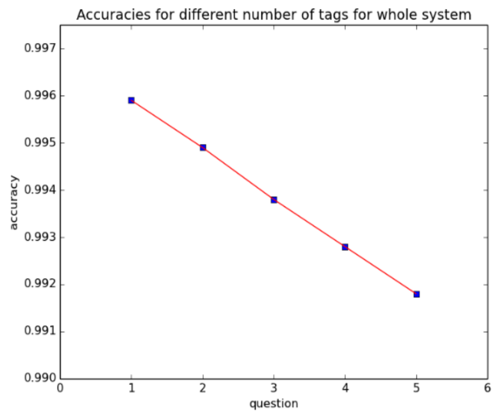


Fig.6. Accuracies of full system for TagStack system with the variation of tags

VI. CONCLUSIONS AND FUTURE WORK

This work presents TagStack system to predict tags on SO. TagStack system uses features extracted title and body the posted questions to predict tags. TagStack system finds tf-idf score and gives top n tags (n=10) using feedback based Naïve Bayes classifier. We presented results on real world publicly available dataset. Results shows that TagStack system is effective in predicting tags.

In the future, we plan to experiment our TagStack system on various Q&A based communities. We will look for better ideas to increase accuracy of theTagStack system. We also plan to evaluate other different classifiers for the improving TagStack system. We would also like to increase our dataset size in future.

REFERENCES

[1] Xia, X., Lo, D., Wang, X., & Zhou, B. (2013). Tag recommendation in software information sites. 2013 10th Working Conference on Mining Software Repositories (MSR).

[2] Denzil Correa , Ashish Sureka, Fit or unfit: analysis and prediction of 'closed questions' on stack overflow, Proceedings of the first ACM conference on Online social networks, October 07-08, 2013, Boston, Massachusetts, USA

[3] Marek Lipczak , Evangelos Milios, Learning in efficient tag recommendation, Proceedings of the fourth ACM conference on Recommender systems, September 26-30, 2010, Barcelona, Spain

[4] Manning, C.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

[5] Firan, C.S., Nejdl, W., Paiu, R.: The Benefit of Using Tag-Based Profiles. In: Proceedings of the Latin American Web Conference, pp. 32–41. IEEE Computer Society, Washington, DC, USA (2007).

[6] E. G. Lezina and A. M. Kuznetsov, "Predict closed questions on stackoverflow," in Proceedings of the Spring Researchers Colloquium on Database and Information Systems, 2013, pp. 10--14.

[7] Treude, C., & Storey, M.-A. (2009). How tagging helps bridge the gap between social and technical aspects in software development. 2009 IEEE 31st International Conference on Software Engineering.

[8] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In Proceedings of Collaborative Web Tagging Workshop at 15th International World Wide Web Conference, 2006.

[9] Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., & Hartmann, B. (2011). Design lessons from the fastest q&a site in the west. Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11.

[10] Zhang, M.-L., & Zhang, K. (2010). Multi-label learning by exploiting label dependency. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '10.

[11] Asaduzzaman, M., Mashiyat, A. S., Roy, C. K., & Schneider, K. A. (2013). Answering questions about unanswered questions of Stack Overflow. 2013 10th Working Conference on Mining Software Repositories (MSR).

[12] Eytan Bakshy , Jake M. Hofman , Winter A. Mason , Duncan J. Watts, Everyone's an influencer: quantifying influence on twitter, Proceedings of the fourth ACM international conference on Web search and data mining, February 09-12, 2011, Hong Kong, China.

[13] D. Correa, A. Sureka, "Chaff from the Wheat: Characterization and Modeling of Deleted Questions on Stack Overflow", *Proceedings of WWW 2014 (23rd international conference on World Wide Web*, 2014.

[14] C. Lee, F. Gutierrez and D. Dou, "Calculating Feature Weights in Naive Bayes with Kullback-Leibler Measure," *2011 IEEE 11th International Conference on Data Mining*, Vancouver,BC, 2011, pp. 1146-1151.

[15] MetaStackOveflow: http://meta.stackexchange.com/questions/139088/how-can-i-determine-a-suitable-tag-for-my-question-on-stack-overflow?lq=1